



TITLE:

円内接多角形問題について : 半径公式再論 (数式処理とその周辺分野の研究)

AUTHOR(S):

森継, 修一

CITATION:

森継, 修一. 円内接多角形問題について : 半径公式再論 (数式処理とその周辺分野の研究). 数理解析研究所講究録 2017, 2054: 153-161

ISSUE DATE:

2017-10

URL:

<http://hdl.handle.net/2433/237157>

RIGHT:

円内接多角形問題について – 半径公式再論*

森継 修一

SHUICHI MORITSUGU†

筑波大学図書館情報メディア系

FACULTY OF LIBRARY, INFORMATION AND MEDIA SCIENCE, UNIVERSITY OF TSUKUBA

1 はじめに

円内接多角形問題とは,

「円に内接する n 角形の各辺の長さ a_1, a_2, \dots, a_n が与えられたとき,
その多角形の面積および外接円の半径 (さらにそれらの関係) を a_1, a_2, \dots, a_n の式で表せ。」

という古典的な幾何学の問題である. 本稿では, 2014 年 12 月の研究集会 [5] 以降の研究の進捗に関して, 特に以下の 3 点について報告する.

- 半径公式 ($n=6$) の計算の効率化
- 半径公式 ($n=7$) の基本対称式表現への変換
- 半径公式 ($n=8$) の計算可能性

数学史上は, $n=3, 4$ が古典的問題, $n \geq 5$ が現代的な問題となり, その間に未解決状態が約 1,300 年続いた. 順に追っていくと, 三角形 ($n=3$) の場合には Heron の公式 (1 世紀) とよばれる結果が知られていて, 外接円の半径 R と三角形の面積 S がそれぞれ,

$$R = \frac{a_1 a_2 a_3}{\sqrt{(a_1 + a_2 + a_3)(-a_1 + a_2 + a_3)(a_1 - a_2 + a_3)(a_1 + a_2 - a_3)}} \quad (1)$$

$$S = \frac{1}{4} \sqrt{(a_1 + a_2 + a_3)(-a_1 + a_2 + a_3)(a_1 - a_2 + a_3)(a_1 + a_2 - a_3)} \quad (2)$$

と表される.

また, 四角形 ($n=4$) に対する結果は, Brahmagupta の公式 (7 世紀) とよばれ, 半径および面積が

$$R = \sqrt{\frac{(a_1 a_2 + a_3 a_4)(a_1 a_3 + a_2 a_4)(a_1 a_4 + a_2 a_3)}{(-a_1 + a_2 + a_3 + a_4)(a_1 - a_2 + a_3 + a_4)(a_1 + a_2 - a_3 + a_4)(a_1 + a_2 + a_3 - a_4)}} \quad (3)$$

$$S = \frac{\sqrt{(-a_1 + a_2 + a_3 + a_4)(a_1 - a_2 + a_3 + a_4)(a_1 + a_2 - a_3 + a_4)(a_1 + a_2 + a_3 - a_4)}}{4} \quad (4)$$

*本研究は科研費 (25330006) の助成を受けたものである.

†moritsug@slis.tsukuba.ac.jp

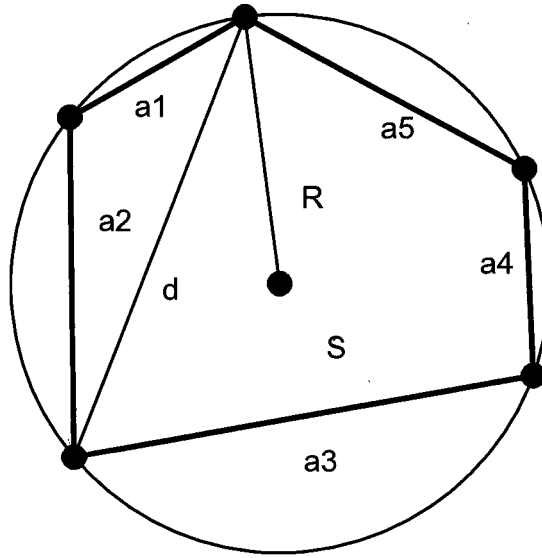


図 1: 円内接五角形の面積 S および外接円半径 R

と表される。ただし、上記の式 (3)(4) は、凸な四角形の場合のみに対応している。 $n = 4$ の場合は、凸でない四角形に対する公式

$$R = \sqrt{\frac{-(a_1a_2 - a_3a_4)(a_1a_3 - a_2a_4)(a_1a_4 - a_2a_3)}{(a_1 + a_2 + a_3 + a_4)(a_1 + a_2 - a_3 - a_4)(a_1 - a_2 - a_3 + a_4)(-a_1 + a_2 - a_3 + a_4)}} \quad (5)$$

$$S = \frac{\sqrt{(a_1 + a_2 + a_3 + a_4)(a_1 + a_2 - a_3 - a_4)(a_1 - a_2 - a_3 + a_4)(-a_1 + a_2 - a_3 + a_4)}}{4} \quad (6)$$

も存在することに注意を要する。

五角形以上に関しては、17 世紀の和算家による五角形の半径公式の計算 [3] を除いて、20 世紀末になつてようやく各公式が明らかにされてきた (表 1)。この表では、第一発見者だけでなく、導出法が異なるものも含めて列挙した。

上記の Heron, Brahmagupta の公式において、面積公式と半径公式から根号部分を消去すると、 $z = 4SR$ または $Z = z^2$ に関する多項式表現が得られ、これを表中では“統合公式”[4] [6] と呼んでいる。この公式は、面積 S と外接円半径 R の関係を表す式とみなすことができるが、Brahmagupta 以来、 $n \geq 5$ に対しては“missing” [7] とされてきたものであり、2015 年の時点に至り、 $n = 5, 6$ に対するすべての公式とその間の関係がすべて解明されたことになる。

面積および半径公式の次数については、Robbins[8] の予想 (のちに定理) により、

$$k_m := \frac{2m+1}{2} \binom{2m}{m} - 2^{2m-1} = \sum_{j=0}^{m-1} (m-j) \binom{2m+1}{j} \quad (7)$$

n	面積公式	半径公式	統合公式
5	Robbins(1994) Pech(2006) Moritsugu(2015)	建部賢弘(1683) 井関知辰(1690) Robbins(1994) Pech(2006)	Moritsugu(2014) Moritsugu(2015)
6	Robbins(1994) Moritsugu(2015)	Moritsugu(2011)	Moritsugu(2015)
7	Maley et al.(2005)	Moritsugu(2011)	(未解決)
8	Maley et al.(2005)	(未解決)	(未解決)

表 1: 円内接 n 角形に対する各公式の計算

すなわち, $d_1, d_2, d_3, \dots = 1, 7, 38, 187, 874, \dots$ とおいたとき, $(2n+1)$ 角形に対して d_n 次, $(2n+2)$ 角形に対して $2d_n$ 次 ($=d_n$ 次 $\times d_n$ 次 に因数分解される) であることが証明されている。すなわち, 五角形に対しては 7 次式, 六角形に対しては 7 次式 \times 7 次式という構造をもち, これが面積公式 \cdot 半径公式 \cdot 統合公式のそれぞれについて成立していることを具体的に示したのが 2014 年 12 月の研究集会での報告 [5] である。

2 半径公式 ($n = 6$) の計算効率化

$n = 3, 4$ における半径公式 ($y = R^2$) の基本対称式表現は, $s_1 = a_1^2 + \dots + a_n^2$, \dots , $s_n = a_1^2 \dots a_n^2$, $\sqrt{s_n} = a_1 \dots a_n$ により

$$(-s_1^2 + 4s_2 + \varepsilon \cdot 8\sqrt{s_4})y - (s_3 + \varepsilon \cdot s_1\sqrt{s_4}) = 0 \quad (8)$$

で与えられる。ここで, crossing parity ε の符号を $\varepsilon = 0$ (三角形), $\varepsilon = 1$ (凸四角形), $\varepsilon = -1$ (非凸四角形) で定めることにより,

$$\begin{cases} F_3(s_i; y) &= (-s_1^2 + 4s_2)y - s_3 \\ F_4^{(+)}(s_i; y) &= (-s_1^2 + 4s_2 + 8\sqrt{s_4})y - (s_3 + s_1\sqrt{s_4}) \\ F_4^{(-)}(s_i; y) &= (-s_1^2 + 4s_2 - 8\sqrt{s_4})y - (s_3 - s_1\sqrt{s_4}) \end{cases} \quad (9)$$

の 3 つの多項式を同時に表している。(辺長でいえば, それぞれ $a_4 = 0$, $a_4 = +a_4$, $a_4 = -a_4$ にとることに相当する。) 五角形・六角形についても, 公式の構造としては $F_5(s_i; y)$, $F_6^{(+)}(s_i; y)$, $F_6^{(-)}(s_i; y)$ が同様の関係を持つことに注意する。

以前の論文 [2] では, 「 $(n+1)$ 角形を対角線 d により n 角形と三角形に分割する」という方針に基づく漸化式を利用して, $n = 6, 7$ の半径公式を求めた。六角形の場合には, 「五角形 + 三角形」に分割したうえで, 対角線 d を終結式により消去する。その概略を抜粋した下記の計算式において, $b_i := a_i^2$, $v := d^2$ という置き換えを行い, 計算の効率化を図っている。これは, $n = 3, 5$ の半径公式において辺長 a_i の偶数乗しか現れないことを利用したものであった。

$$\begin{cases} h_6(b_1, \dots, b_6; y) &:= \text{Res}_v(h_5(b_1, b_2, b_3, b_4, v; y), h_3(b_5, b_6, v; y))/y^8 \\ \Phi_6(a_1, \dots, a_6; y) &:= h_6(a_1^2, \dots, a_6^2; y) \\ &= B_{14}y^{14} + \dots + B_1y + B_0 = 0 \quad (B_i \in \mathbb{Z}[a_1, \dots, a_6]) \end{cases} \quad (10)$$

最後に, $\Phi_6(a_i; y)$ (497,417 項) を因数分解することにより

$$\Phi_6(a_i; y) = \phi(a_i; y) \cdot \varphi(a_i; y) \quad (\deg_y \phi = \deg_y \varphi = 7) \quad (11)$$

として、約9時間のCPU時間で各19,449項の因子を得ている(当時の計算環境: Maple14(Win64), Xeon(2.93 GHz) × 2, 24GB RAM). しかしながら、これらは $\phi(a_1, \dots, a_7, a_8; y) = \varphi(a_1, \dots, a_7, -a_8; y)$ という関係にあるので、凸六角形に対応する因子 $\varphi(a_i; y) = F_6^{(+)}(s_i; y)$ のみを求めれば十分である。

したがって、新たな方針として「2つの凸四角形に分割し対角線 d を消去」とすると、

$$\varphi(a_i; y) = \text{Res}_d(h_4^{(+)}(a_1, a_2, a_3, d; y), h_4^{(+)}(a_4, a_5, a_6, d; y))/y \quad (12)$$

と求められ、因数分解は不要でCPU時間は1秒未満となった。なお、 $\text{Res}_d(h_4^{(+)}, h_4^{(-)})$ および $\text{Res}_d(h_4^{(-)}, h_4^{(+)})$ からは $\phi(a_i; y) = F_6^{(-)}(s_i; y)$ に対応する因子が、 $\text{Res}_d(h_4^{(-)}, h_4^{(-)})$ からは $\varphi(a_i; y) = F_6^{(+)}(s_i; y)$ が得られるので、これらのうち1組だけ計算を実行すれば十分である。

ただし、 $n=7$ の場合には、論文 [2] で用いた「七角形=六角形(14次式)+三角形」という分割以外の方法では計算に成功していない。その他の分割で「凸六角形(7次式)+三角形」「五角形(7次式)+凸四角形」という組み合わせも試したが、終結式の展開・整理が困難のようである。これは、漸化式(10)を利用した場合、上記と同様に $b_i := a_i^2$ の置き換えによって式が簡単化できることが効いていると思われる。

3 半径公式 ($n=7$) の基本対称式表現への変換

3.1 基本対称式表現への変換の効率化

これまでの計算方法では、 n 変数の基本対称式からなるイデアル

$$I = \{s_1^{(n)} - (a_1^2 + \dots + a_n^2), \dots, s_n^{(n)} - (a_1^2 \dots a_n^2)\} \quad (13)$$

に対して、グループ順序 (Maple では “lexdeg”) の Gröbner 基底

$$G := \text{Basis}(I, \{a_1, \dots, a_n\} \succ \{s_1, \dots, s_n\}) \quad (14)$$

を求め、簡約化の組み込み関数を用いて $p := \text{NormalForm}(f, G)$ としていた。この方法では、6 変数の場合 ($n=6$ に対する面積・半径・統合公式) の変換には成功していたが、7 変数になると全く機能しなくなった。一例としては、 $a_1^{20} \dots a_7^{20} \Rightarrow s_7^{10}$ の計算がいつまでたっても終わらず、NormalForm 関数の内部での簡約順序に任せるのは現実的ではなくなった。

したがって、グレブナー基底による簡約手順を以下のように構成し直すこととした。式の簡単化のため $b_i := a_i^2$ と置き換えたうえで、純辞書式順序 $b_1 \succ \dots \succ b_n \succ s_1 \succ \dots \succ s_n$ で計算したイデアル I の Gröbner 基底 $G = \{g_1, \dots, g_n\}$ を求めると、各基底多項式の主項が b_1, b_2^2, \dots, b_n^n となることを利用する。 $n=3$ の例で示すと、

$$\begin{cases} g_1 &= b_1 + (b_2 + b_3 - s_1) \\ g_2 &= b_2^2 + (b_3 - s_1)b_2 + (b_3^2 - b_3s_1 + s_2) \\ g_3 &= b_3^3 - s_1b_3^2 + s_2b_3 - s_3 \end{cases} \quad (15)$$

という形をしているので、 $p := \text{NormalForm}(f, G)$ を求めるには、多項式剰余を求める関数 Rem を用いて

$$\begin{cases} r_1 &:= \text{Rem}(f, g_1; b_1) \\ r_2 &:= \text{Rem}(r_1, g_2; b_2) \\ p &:= \text{Rem}(r_2, g_3; b_3) \end{cases} \quad (16)$$

と多項式の除算を3回を実行すれば、 b_1, b_2, b_3 が順次消去されて s_i だけの式が得られる。(一般の場合は、これを拡張して n 回の剰余計算を行う。)

この計算方法により、半径公式 ($n = 7$) の基本対称式表現を求めることに今回初めて成功した。具体的な式の形は、辺長 a_i による表現

$$F_7(y) = B_{38}y^{38} + \cdots + B_1y + B_0 \quad (B_i \in \mathbf{Z}[a_1^2, \dots, a_7^2]; \quad 337,550,051 \text{ 項}) \quad (17)$$

から変換して、

$$\tilde{F}_7(y) = \tilde{B}_{38}y^{38} + \cdots + \tilde{B}_1y + \tilde{B}_0 \quad (\tilde{B}_i \in \mathbf{Z}[s_1, \dots, s_7]; \quad 199,695 \text{ 項}) \quad (18)$$

となり、今回の使用環境 (Maple15, Win64, Xeon(2.93GHz)×2, 192GB RAM) では、合計約 87 時間を要した。ちなみに、面積公式 ($n = 7$) は [1] にしたがって計算すると

$$\tilde{\Phi}_7(x) = x^{38} + \tilde{C}_{37}x^{37} + \cdots + \tilde{C}_1x + \tilde{C}_0 \quad (x = (4S)^2; \quad \tilde{C}_i \in \mathbf{Z}[s_1, \dots, s_7]; \quad 955,641 \text{ 項}) \quad (19)$$

という形で表される。

3.2 コーシーの方法によるメモリ使用量の節約

上述の方法を 8 変数の場合に適用すると、剰余計算における中間表式膨張が著しく、メモリ不足ですぐに計算不能に陥った。そこでコーシーの方法とよばれる関係式を用いて、「 n 変数の問題を多数の $n-1$ 変数の問題に分割する」計算法を試した。

x_1, \dots, x_n を変数とする k 次の基本対称式を s_k 、 x_2, \dots, x_n を変数とする k 次の基本対称式を t_k とおくと、これらの間に次の関係が成り立つ。

$$\begin{cases} s_1 &= x_1 + t_1 \\ s_2 &= t_1x_1 + t_2 \\ \cdots &\cdots \cdots \\ s_{n-1} &= t_{n-2}x_1 + t_{n-1} \\ s_n &= t_{n-1}x_1 \end{cases} \quad (20)$$

上から順に「 t_i について解いて、次の式に代入する」ことを第 $(n-1)$ 行まで繰り返すと、

$$\begin{cases} t_1 &= s_1 - x_1 \\ t_2 &= s_2 - t_1x_1 = s_2 - s_1x_1 + x_1^2 \\ \cdots &\cdots \cdots \\ t_{n-1} &= s_{n-1} - t_{n-2}x_1 = s_{n-1} - \cdots + (-1)^{n-1}x_1^{n-1} \end{cases} \quad (21)$$

となり、 t_1, \dots, t_{n-1} は x_1, s_1, \dots, s_{n-1} の多項式で表される。さらに第 n 行に代入すれば

$$g(x_1) = (-1)^{n-1}x_1^n + \cdots + s_{n-1}x_1 - s_n = 0 \quad (22)$$

という関係式が得られる。これらを用いて、以下の手順で n 変数の基本対称式表現を求める。

与えられた n 変数の対称式を x_1 について整理して

$$f(x_1, \dots, x_n) = a_\ell(x_2, \dots, x_n)x_1^\ell + \cdots + a_1(x_2, \dots, x_n)x_1 + a_0(x_2, \dots, x_n) \quad (23)$$

とすると、各係数 $a_j(x_2, \dots, x_n)$ は x_2, \dots, x_n の対称式となる。多項式の除算の繰り返しのより、前節の方法でこれらを基本対称式 t_k による表現に変換すると、

$$f(x_1, \dots, x_n) = \tilde{a}_\ell(t_1, \dots, t_{n-1})x_1^\ell + \cdots + \tilde{a}_1(t_1, \dots, t_{n-1})x_1 + \tilde{a}_0(t_1, \dots, t_{n-1}) \quad (24)$$

となる。これは、「 n 変数の問題 $\times 1$ 個」を「 $(n-1)$ 変数の問題 $\times \ell$ 個」に分割したことに相当する。

この結果に式 (21) を適用すると、 t_k は x_1, s_k の表現に置き換わり、あらためて x_1 について整理すると

$$f(x_1, \dots, x_n) = c_m(s_1, \dots, s_{n-1})x_1^m + \dots + c_1(s_1, \dots, s_{n-1})x_1 + c_0(s_1, \dots, s_{n-1}) \quad (25)$$

が得られる。最後に式 (22) の $g(x_1)$ により x_1 についての剰余を計算すれば、 x_1 が完全に消去されて、 s_1, \dots, s_n による表現が得られることになる。

次節に示す $n=8$ の場合の計算は、すべてこの方法によって結果を得ている。ただし、 $n=7$ の式 (18) にも適用してみたところ、合計 CPU 時間は約 103 時間と却って増えてしまった。これは、問題を分割した結果、6 変数の問題を多数計算する必要が生じたためで、コーシーの方法はメモリの節約には効果があるものの、計算時間の短縮には必ずしもつながらないケースが存在することを示している。

4 半径公式 ($n=8$) の計算可能性

y の次数	a_i 表現の項数	s_i 表現の項数	全次数
0	5,554,128	918	70
1	13,298,304	1,870	69
2	26,940,233	3,432	68
3	48,012,824	5,732	67
4	77,750,132	8,931	66
5	114,947,440	12,670	65
6	158,302,913	17,129	64
7	204,390,480	21,592	63
8	250,654,676	26,179	62
9	293,931,056	30,200	61
32	67,463,040	11,436	38
33	56,784,240	10,026	37
34	47,533,327	8,743	36
35	39,574,496	7,514	35
36	32,771,272	6,385	34
37	26,990,336	5,260	33
38	22,105,457	4,231	32

表 2: 半径公式 ($n=8$) における各係数の特徴

八角形の外接円半径を求めるため、「八角形＝六角形＋四角形」に分割し、対角線 d に関する多項式に整理する。六角形・四角形とも凸な場合の式を用いて

$$\begin{cases} F_6^{(+)}(y) = y^7 d^{16} - a_1 a_2 a_3 a_4 a_5 y^5 d^{15} + u_{14} d^{14} + \dots u_1 d + u_0 \\ \quad (19,449 \text{ 項 : } 14 \text{ 次以下の項の係数を別変数 } u_i \text{ に置き換え}) \\ F_4^{(+)}(y) = y d^4 + a_6 a_7 a_8 d^3 + \dots \end{cases} \quad (26)$$

(19 項)

とおく。ここから d についての終結式を計算すると、 y についての最低次項が y^2 になることから、この因子を除くとして

$$Q(y) := \text{Res}_d(F_6^{(+)}(y), F_4^{(+)}(y))/y^2 \quad (27)$$

とおく。これは、 y について 42 次式で、置き換えた変数 u_i による表現のままで数えて 9,576,487 項をもつ。部分的に u_i を展開すると y^0, y^1, y^2, y^3 の各係数 = 0 が確かめられるので、残りの y^4, \dots, y^{42} の各係数を順次取り出して u_i を展開することにより、凸な場合の八角形の外接円半径の公式

$$F_8^{(+)}(y) := Q(y)/y^4 = C_{38}y^{38} + \dots + C_1y + C_0 \quad (28)$$

が(原理的には)求められる。現時点で結果が確認できた係数の特徴を表 2 に示す。より項数の多い C_{10} から C_{31} についての計算は未完である。なお、「全次数」の欄については、次節で議論する。

実際のプログラムとしては、各係数の計算のため、 a_1, \dots, a_5, y の多項式である u_i による表現から特定の y^k の係数部分を抜き出し、式のサイズが爆発しないようできるだけ細かい部分に小分けして展開を行う。さらに辺長 a_i による表現が求められた後、前節の「コーシーの方法」により、8 変数の基本対称式表現への変換を行った。

使用環境は Maple18(Linux), Xeon(8 core, 2.6GHz) \times 2, 256GB であるが、メモリ使用量を抑えることで、結果的に(時間さえかければ)計算が可能になったと考えられる。例えば、 C_9 では、56 個のジョブに分割して、CPU 時間の累計は約 300 日である。素朴なプログラミングでは、メモリの使用率が上がるとガーベージコレクションにばかり CPU 時間が取られてしまう現象が見られたため、今回採用した「問題分割による工夫」は、一定の成果を上げていると考えられる。ただし、残った係数はさらに巨大になるため、メモリ不足を起こす可能性もあり、実行できたとしても全係数についての計算には数十年単位の CPU 時間が必要と見込まれる。

結果については、次の 2 方法により $n = 7$ の場合の拡張になっていることを確認している。

検算 (1) $a_8 := 0$ を代入して、 $F_7(y)$ の係数と一致するかどうか。

検算 (2) 基本対称式表現に変換後、 $\sqrt{s_8} := 0$ を代入して、 $\tilde{F}_7(y)$ の係数と一致するかどうか。

基本対称式表現に正しく変換されたことと合わせて、これらの検算から、係数が求められた範囲では正しい結果が得られているとみられる。特に、主係数について

$$C_{38} = \prod_{64 \text{ 項}} \left(a_1 + \sum_{j=2}^8 (-1)^{k_j} a_j \right) \quad k_j \in \{0, 1\}, \quad \sum_{j=2}^8 k_j \equiv 1 \pmod{2} \quad (29)$$

という構造をもつこと ($a_1 \pm a_2 \pm \dots \pm a_8$ の形の式のうち、+ が偶数個のものの積) が確かめられ、これは $n = 7$ の場合の自然な拡張になっている。

5 各公式の形状について

Brahmagupta の半径公式 ($n=4$, 凸) を展開した形で示せば、

$$\begin{aligned} & (2(a_1^2a_2^2 + a_1^2a_3^2 + a_1^2a_4^2 + a_2^2a_3^2 + a_2^2a_4^2 + a_3^2a_4^2) + 8a_1a_2a_3a_4 - (a_1^4 + a_2^4 + a_3^4 + a_4^4)) R^2 \\ & - (a_1^2a_2^2a_3^2 + a_1^2a_2^2a_4^2 + a_1^2a_3^2a_4^2 + a_2^2a_3^2a_4^2 + a_1a_2a_3a_4(a_1^2 + a_2^2 + a_3^2 + a_4^2)) = 0 \end{aligned} \quad (30)$$

であり、これを a_i^2 の基本対称式で表したものが、式 (9) の $F_4^{(+)}(s_i; y)$ である。この式において R^2 の係数は同次式でもあり、含まれる単項式の全次数は a_i^2 を単位として 2 である。(例外的に、 $a_1a_2a_3a_4 = \sqrt{s_4}$ の

主変数の次数	1	0
面積公式 ($x = (4S)^2$)	0	2
半径公式 ($y = R^2$)	2	3
統合公式 ($Z = (4SR)^2$)	0	3

表 3: 各公式 ($n = 3, 4$) における各係数の全次数

次数は 2 とする。) 基本対称式表現からも, 「各 s_i の次数は i 」であることに注意して全次数を求めれば, R^2 の係数は 2 次の同次式であることが確認できる. 同様に, 定数項は全次数 3 の単項式のみからなる同次式であることがわかる. この特徴は, 面積公式・統合公式にも共通しており, $n = 3, 4$ の場合をまとめたものが表 3 である. 同様に, $n = 5, 6$ の場合をまとめたものが表 4 であり, 次数が規則的に分布していることがわかる.

主変数の次数	7	6	5	4	3	2	1	0
面積公式 ($x = (4S)^2$)	0	2	4	6	8	10	12	14
半径公式 ($y = R^2$)	8	9	10	11	12	13	14	15
統合公式 ($Z = (4SR)^2$)	0	3	6	9	12	15	18	21

表 4: 各公式 ($n = 5, 6$) における各係数の全次数

$n = 7$ の場合について, 面積公式・半径公式から次数を確認したものが表 5 である. 統合公式 ($n = 7$) については, 定数項と Z^{38} の項のみ理論的に求められるが, その他の項は未知なので, 規則的に分布していると仮定した予想値を示してある.

主変数の次数	38	37	36	2	1	0
面積公式 ($x = (4S)^2$)	0	2	4	72	74	76
半径公式 ($y = R^2$)	32	33	34	68	69	70
統合公式 ($Z = (4SR)^2$)	0	(3)	(6)	(108)	(111)	114

表 5: 各公式 ($n = 7$) における各係数の全次数 (カッコ内は予想値)

面積公式 ($n = 8$) は既知なので, 表 5 の値に一致することが確認済みである. 現在計算中の半径公式 ($n = 8$) については, 表 2 の「全次数」の欄に示したとおり, 計算済みの係数については $n = 7$ の場合と一致していることが確認でき, これまでの計算結果が正しいことを間接的に裏付けている.

6 まとめと今後の課題

本稿では, 円内接多角形問題に関する研究について, 以下の点における進展をまとめて示した.

- (1) 半径公式 ($n = 6$) の計算法を見直し, 巨大な多項式の因数分解を回避することにより, ほぼ瞬時に計算可能であることを示した. ただし, $n = 7$ に対しては同様の見直しが効かず, 以前の論文 [2] 以外の方法では, 計算に成功しなかった.

- (2) 半径公式 ($n = 7$) の基本対称式表現の計算に初めて成功した。辞書式順序のグレブナー基底により順次簡約する方法に加え、コーシーの方法によって $(n - 1)$ 変数の問題に分割するアルゴリズムを実装し比較した。後者は $n = 8$ の場合に特に有効であった。
- (3) 半径公式 ($n = 8$) のいくつかの係数の計算に成功し、辺長 (a_i) による表現に加え、基本対称式 (s_i) による表現まで求められた。数 GB 単位の巨大な式を展開・整理する必要があるため、内部のデータ構造まで考慮したプログラミングの方向性は見えたが、数式処理システムの性能を限界まで引き出すのは依然困難である。
- (4) これまでに求められた各公式の「係数における単項式の全次数」を調べ、どの場合も規則的に分布していることが確かめられた。これらの値の持つ意味については今後の検討課題であるが、「数値補間により未知の係数を求める算法」に適用できる可能性が期待される。

今後の遠い目標としては、 $z = 4SR$, $Z = z^2$ に関する統合公式 ($n = 7, 8$) の導出が考えられるが、現状の延長では実行困難である。面積公式 ($n = 7, 8$) の導出については Maley 他 [1] による解析があるが、半径公式・統合公式に関しては、このような理論的整備は未知であるとみられる。

参 考 文 献

- [1] Maley, F. M., Robbins, D. P., and Roskies, J.: On the Areas of Cyclic and Semicyclic Polygons, *Advances in Applied Mathematics*, **34**(4), 2005, 669–689.
- [2] Moritsugu, S.: Computing Explicit Formulae for the Radius of Cyclic Hexagons and Heptagons, *Bulletin of Japan Soc. Symbolic and Algebraic Computation*, **18**(1), 2011, 3–9.
- [3] 森継修一: 円内接多角形問題と「算法発揮 (1690)」における解について, 京都大学数理解析研究所講究録, **1815**, 2012, 124–132.
- [4] 森継修一: 円内接多角形問題について —半径公式と面積公式の統合—, 京都大学数理解析研究所講究録, **1907**, 2014, 174–181.
- [5] 森継修一: 円内接多角形問題における面積公式・半径公式・統合公式について, 京都大学数理解析研究所講究録, **1955**, 2015, 91–101.
- [6] Moritsugu, S.: Integrated Circumradius and Area Formulae for Cyclic Pentagons and Hexagons, *ADG 2014* (Botana, F. and Quaresma, P., eds.), *LNAI*, **9201**, Springer, 2015, 94–107.
- [7] Pech, P.: Computations of the Area and Radius of Cyclic Polygons Given by the Lengths of Sides, *ADG2004* (Hong, H. and Wang, D., eds.), *LNAI*, **3763**, Gainesville, Springer, 2006, 44–58.
- [8] Robbins, D. P.: Areas of Polygons Inscribed in a Circle, *Discrete & Computational Geometry*, **12**(1), 1994, 223–236.